CONNEXIONS™

The Interoperability Report

December 1987 Volume 1, No. 8

ConneXions - The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

From the Editor

This issue is being released at the 2nd TCP/IP Interoperability Conference. As an attendee, you will receive a year's subscription to *ConneXions* at no additional cost.

Douglas Comer is currently putting the finishing touches on his book "Internetworking With TCP/IP Principles, Protocols, and Architecture". This month we bring you an article on address mapping taken from the book.

The Internet Activities Board (IAB) coordinates research and operations in the DARPA/NSF Internet community. We asked Dr. Jon Postel of USC-ISI to give us an overview of this board.

It is not our normal policy to reprint RFCs in *ConneXions*, but since it is almost "that time of the year", we decided to end Volume 1 with RFC 968 by Vint Cerf, entitled "'Twas the Night Before Start-up".

Advanced Computing Environments has put together a set of 9 tutorials which will be held from April 25th through 27th, 1988 at the Hyatt Regency Crystal City in Arlington, VA. The tutorials are:

Introduction to TCP/IP	Doug Comer, Purdue University	(1 day)
TCP/IP In-Depth	Len Bosack, cisco Systems	(2 days)
Network Security	Steve Walker/Curt Barker, Trusted Information Systems	(1 day)
OSI Reference Model and Protocols	Hal Folts, The Omnicom Institute	(1 day)
Building Distributed Applications in an OSI Framework	Marshall Rose, The Wollongong Group	(1 day)
Local Area Networks	Charles Brown, Complete Systems	(2 days)
Berkeley UNIX Networking	Mike Karels, UC Berkeley	(2 days)
TCP/IP for the VM Systems Programmer	Nick Gimbrone, Cornell University	(2 days)
Microcomputer Networking with TCP/IP	David Crocker, The Wollongong Group	(1 day)

For more information call us at 408-996-2042.

ConneXions is published by Advanced Computing Environments, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. Phone: 415-941-3399.

ISSN 0894-5926

Mapping Internet Addresses to Ethernet Addresses

by Douglas Comer, Purdue University

The Internet address scheme assigns a 32-bit IP address to each host and uses the IP address to route traffic. Ultimately, Internet software must map the IP address into an address that the underlying network hardware understands. How can a host or a gateway map an IP address into the correct physical hardware address given only an IP address? This article considers that mapping, showing how it is implemented for the two most common physical network address schemes.

The address resolution problem

Consider two machines A and B that share a physical network. Each has an assigned Internet address IA and IB, and a physical hardware address PA and PB. The goal is to devise low-level software that hides physical addresses and allows higher-level programs to work only with Internet addresses. The ultimate communication, however, must be carried out by physical networks using whatever physical addressing schemes the hardware supplies. So the question arises, suppose machine A wants to send a packet to machine B across a physical network to which they both attach, but has only B's Internet address IB, how does it map that address to B's physical address, PB?

The problem of mapping Internet addresses to physical addresses is known as the *address resolution problem*, and has been solved in several ways. Some internet designs keep tables in each machine as pairs of internet and physical addresses. Others solve the problem by encoding hardware addresses in the internet addresses. Using either approach exclusively makes internet addressing awkward at best.

Two types of physical addresses

There are two basic types of physical addresses, exemplified by the Ethernet, which has large, fixed physical addresses, and the proNET-10*, which has small, compact, easily changed physical addresses. Address resolution is difficult for Ethernet-like networks, but easy for networks like proNET-10. We will consider the easy case first.

Resolution through direct mapping

Consider a proNET-10 token passing ring network. It uses small integers for physical addresses, and allows the customer to choose a hardware address when installing an interface board in a computer. The key to making address resolution easy for a proNET network lies in observing that as long as one has the freedom to choose both Internet and physical addresses, they can be selected such that parts of them are the same. Typically, one assigns Internet addresses with the host id portion equal to 1, 2, 3, and so on, and then, when installing network interface hardware, selects a physical address that corresponds to the Internet address. For example, one would select physical address 3 for a machine with the Internet address 192.5.48.3 because 192.5.48.3 is a class C address with the host portion equal to 3.

*proNET-10 is the name of a commercial network hardware product manufactured by Proteon Corporation.

For networks like proNET-10, computing a physical address from an Internet address is trivial. The computation consists of extracting the low order byte of the Internet address. It is computationally efficient because it requires only a few machine instructions. It is easy to maintain because the mapping can be performed without reference to external data. Finally, new machines can be added to the network without changing data or recompiling code.

Resolution through dynamic binding

To understand why address resolution is difficult for some networks, consider the Ethernet. The Ethernet has 48-bit physical addresses assigned by vendors when they manufacture interface boards. As a consequence, replacing a board that fails changes the machine's physical address. Furthermore, because the Ethernet address is 48 bits long, there is no hope it could be encoded in a 32-bit Internet address.

ARP

Designers of the Internet found a creative solution to the address resolution problem for networks like the Ethernet, a solution that allows new machines to be added to the network without recompiling code, and does not require maintenance of a centralized database. To avoid maintaining a table of mappings, they chose to use a low-level protocol to bind addresses dynamically. Termed the *Address Resolution Protocol* (ARP), the low level protocol provides a mechanism that is both efficient and easy to maintain.

As Figure 1 shows, the idea behind dynamic resolution with ARP is simple: when host A wants to resolve Internet address IB, it broadcasts a special packet that asks the host with Internet address IB to respond with its physical address, PB. All hosts, including B, receive the request, but only host B recognizes its Internet address and sends a reply that contains its physical address.

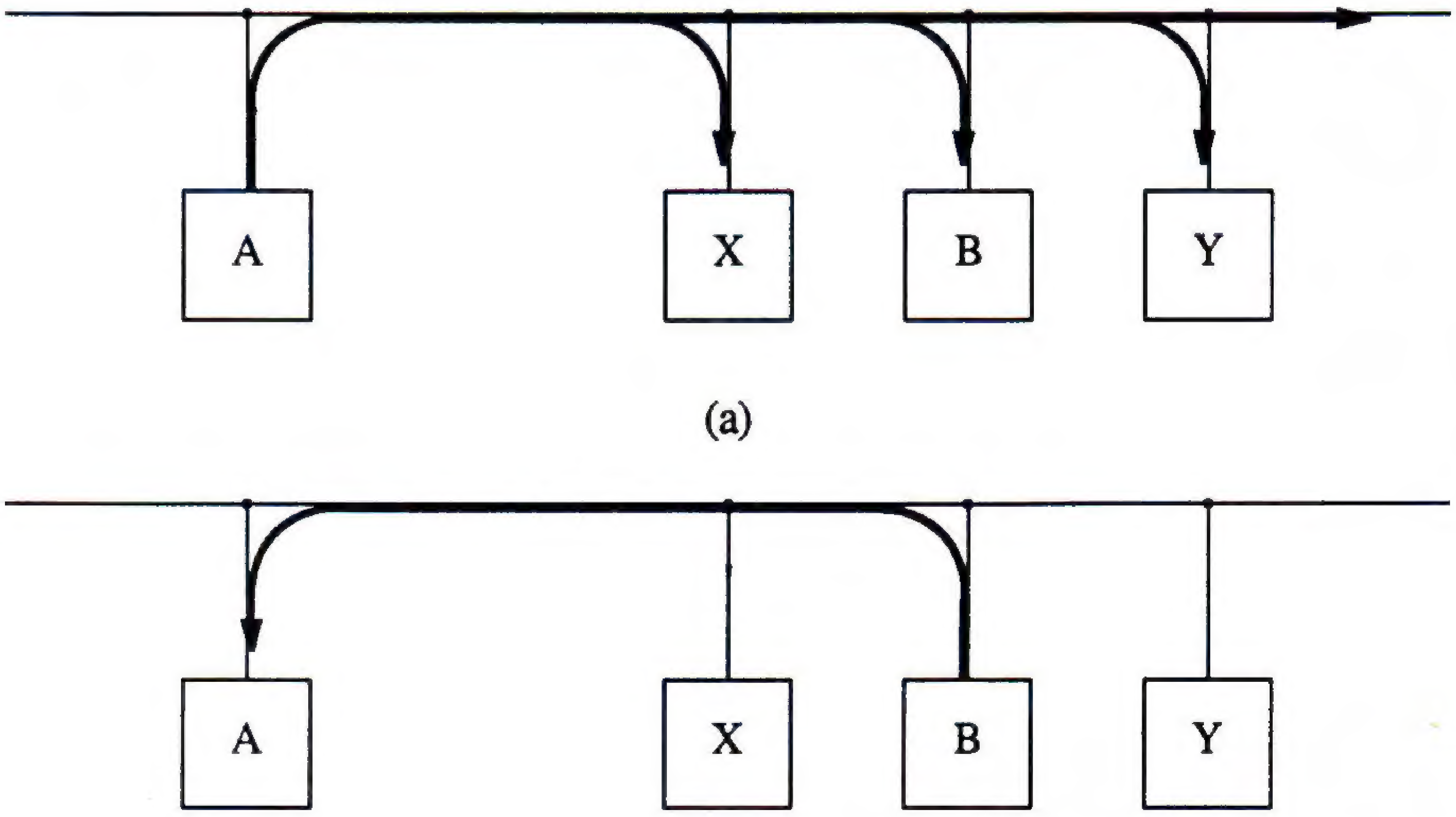


Figure 1: The ARP protocol. To determine PB, B's physical address, from IB, its Internet address, (a) host A broadcasts an ARP request containing IB to all machines, and (b) host B responds with an ARP reply that contains the pair (IB,PB).

continued on next page

Mapping Internet Addresses *(continued)*

When A receives the reply, it learns B's physical hardware address, and uses that address to send the Internet packet directly to B. We can summarize:

> *The Address Resolution Protocol, ARP, allows a host to find the physical address of a target host on the same physical network, given only the target's Internet address.*

The address resolution cache

It may seem silly that for A to send a packet to B it first sends a broadcast that reaches B. Or it may seem even sillier that A broadcasts the question, "how can I reach you?" instead of just broadcasting the packet it wants to deliver. But there is an important reason for the exchange. Broadcasting is far too expensive to be used every time one machine needs to transmit a packet to another. To reduce communication costs, hosts that use ARP maintain a cache of recently acquired Internet-to-physical address bindings so they do not have to use ARP again. Whenever a host receives an ARP reply, it saves the result in its cache for successive lookups. When transmitting a packet, the host always looks in its cache for a binding before sending an ARP request. If the host finds the desired binding in its cache, it need not use the network. Experience shows that because most network communication involves more than one packet transfer, even a small cache is worthwhile.

ARP refinements

Several refinements of ARP are possible. First, observe that if host A is about to use ARP because it needs to send to B, there is a high probability that host B will need to send to A in the near future. If we anticipate B's need, we can avoid extra network traffic by arranging for A to include its Internet-to-physical address binding when sending a request to B. Second, notice that because A broadcasts its initial request, all machines on the network receive it, and can extract and store in their cache A's Internet-to-physical address binding. Third, when a new machine appears on the net (e.g., when an operating system reboots), we can avoid having every other machine run ARP by broadcasting the new pair of internet address and physical address. The following rule summarizes refinements:

> *The sender's internet-to-physical address binding is included in every ARP broadcast; receivers add the internet-to-physical address binding information to their cache before processing an ARP packet.*

Relationship of ARP to other protocols

ARP provides one possible mechanism to map from Internet addresses to physical addresses; we have already seen that network hardware like the proNET-10 do not need it. The point is that ARP would be completely unnecessary if we could make all network interfaces understand their Internet address. Thus, ARP merely imposes a new addressing scheme on top of whatever low-level addressing mechanism the hardware uses. The idea can be summarized:

> *ARP is a low-level protocol that hides the underlying network physical addressing, permitting us to assign Internet addresses of our choosing to every machine. We think of it as part of the physical network system, and not as part of the Internet protocols.*

ARP implementation

Functionally, ARP is divided into two parts. One part determines physical addresses when sending a packet, and the other answers requests from other machines. Address resolution for outgoing packets seems straightforward, but small details complicate an implementation. Given a destination Internet address, the host consults its ARP cache to see if it knows the mapping to physical address. If it does, it extracts the physical address, places the data in a frame using that address, and sends the frame. If it does not know the mapping, it must broadcast an ARP request and wait for a reply.

Broadcasting an ARP request to find an address mapping can become complex. The target machine could be down, or just too busy to accept the request, in which case the sender may not receive a reply, or the reply may be delayed. Or the initial ARP broadcast request could be lost (in which case the sender should retransmit, at least once). Meanwhile, the host must store the original outgoing packet so it can be sent once the address has been resolved. In fact, the host must decide whether to allow multiple outstanding ARP requests (most do not), and if allowing multiple requests, it must take care not to broadcast multiple ARP requests for a given target Internet address. Finally, the cached value could be out of date, making successful transmission impossible.

The second part of the ARP code handles ARP packets that arrive from the network. The handler first copies the sender's Internet and physical addresses into its ARP cache for later use. It then processes the packet.

The host must handle two types of incoming ARP packets. It must examine incoming ARP request packets to see if it is the target of a request from another machine, If so, the ARP handler forms and sends a reply, supplying its physical address. If not, the packet is requesting a mapping for another machine, and can be ignored.

The other interesting case occurs when an ARP reply arrives. The handler always records the sender's Internet-to-physical binding in its cache before processing. If the reply answers a previously issued query, the handler must then arrange to have the waiting outgoing packet(s) sent. If no outgoing packets await the reply, the host simply stops processing the packet.

ARP encapsulation and identification

When ARP messages travel from one machine to another, they must be carried in physical Ethernet packets as Figure 2 shows.

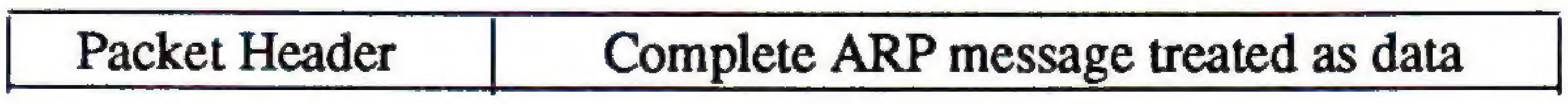


Figure 2: An ARP message encapsulated in an Ethernet packet.

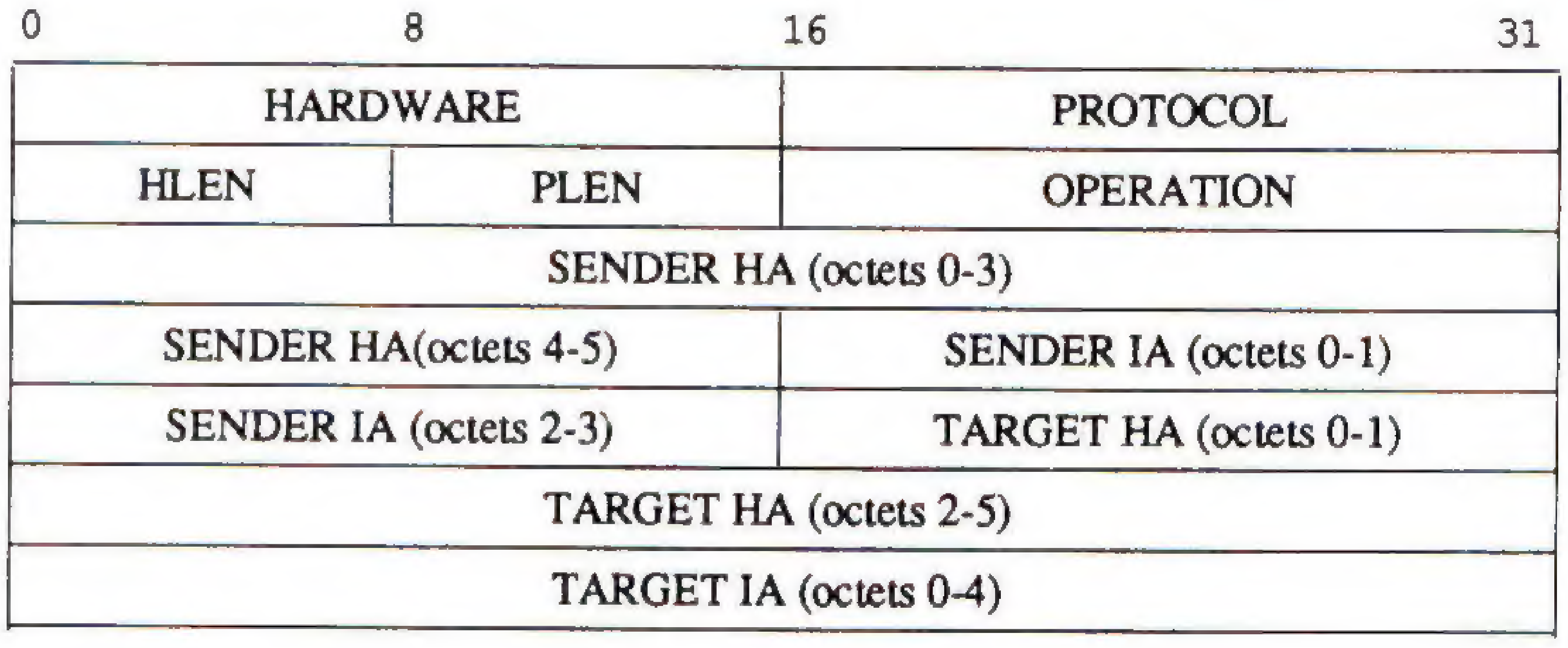
continued on next page

Mapping Internet Addresses *(continued)*

To identify the frame as carrying an ARP request or ARP reply, the sender assigns a special value to the type field in the frame header, and places the ARP message itself in the frame's data field. When a frame arrives at a host, the system examines the frame type to determine what it contains. For example, on an Ethernet, ARP requests have a type field of 0x0806 and replies have a type of 0x8035. These are standard values, assigned by the authority that sets Ethernet standards, and are universally accepted.

ARP protocol format

Unlike most protocols, the data in ARP packets does not have a fixed-format header. Instead, the message is designed to be useful with a variety of network technologies, so early header fields contain counts that specify lengths of succeeding fields. In fact, ARP can be used with arbitrary physical addresses and arbitrary protocol addresses. The example in Figure 3 below shows the 28-octet ARP message format used on Ethernet hardware (where physical addresses are 48-bits or 6 octets long), when resolving DARPA Internet protocol addresses (4 octets long). Unlike most of the Internet protocols, the variable-length fields in ARP packets do not align on 32-bit boundaries, making the diagram difficult to read. For example, the sender's hardware address, labeled SENDER HA, occupies 6 contiguous octets, so it spans two lines in the diagram. Nevertheless, we have chosen this format because it is standard throughout the Internet literature.



0	8	16	31
HARDWARE		PROTOCOL	
HLEN	PLEN	OPERATION	
SENDER HA (octets 0-3)			
SENDER HA(octets 4-5)		SENDER IA (octets 0-1)	
SENDER IA (octets 2-3)		TARGET HA (octets 0-1)	
TARGET HA (octets 2-5)			
TARGET IA (octets 0-4)			

Figure 3: The format of ARP/RARP messages used for Internet-to-Ethernet address resolution.

Field HARDWARE specifies a hardware interface type for which the sender seeks an answer; it is 1 for Ethernet. Field OPERATION specifies an ARP request (1), ARP response (2), RARP* request (3), or RARP response (4).

Fields HLEN and PLEN allow ARP to be used with arbitrary networks because they specify the length of the physical hardware address and the length of the protocol address. The sender supplies its hardware address and Internet address, if known, in fields SENDER HA and SENDER IA.

When making a request, the sender also supplies the target Internet address (ARP), or target hardware address (RARP), using fields TARGET HA and TARGET IA. A response carries both the target machine's hardware and Internet addresses.

*RARP (Reverse Address Resolution Protocol), which uses the same message format, will be described in a later article.

Summary

Internet addresses are assigned independent of the physical hardware address. However, high-level network software that uses Internet addresses must ultimately map them into physical hardware addresses when delivering packets to their final destination. If hardware addresses consist of small integers that can be changed easily, a direct mapping can be established by having the physical address be encoded in the Internet address; otherwise, the mapping must be performed dynamically. The ARP protocol performs dynamic address resolution, using only the low-level network communication system. It permits machines to resolve addresses without keeping a permanent record of bindings.

A machine uses ARP to find the hardware address of another machine by broadcasting an ARP request that contains the Internet address for which it searches. Each machine responds to requests for its physical hardware address, and sends replies that contain the needed binding.

To make ARP efficient, each machine caches Internet-to-physical address bindings. Because Internet traffic tends to consist of a sequence of interactions between pairs of machines, the cache eliminates most ARP broadcast requests.

Additional information

The address resolution protocol used here is given by Plummer [1982], and has become an Internet standard. Dalal and Printis [1981] describe the relationship between Ethernet and Internet addresses, and Clark [1982] discusses addresses and bindings in general. The text by Comer [1987] provides and example implementation of ARP for the Xinu operating system.

References

Clark, D., "Names Addresses, Ports, and Routes," RFC 814, 1982.

Comer, D., Operating System Design Vol 2. Internetworking with Xinu, Prentice-Hall, 1987.

Dalal & Printis, "48-Bit Absolute Internet and Ethernet Host Numbers," Proceedings of the Seventh Data Communication Symposium, 1981.

Plummer, D., "An Ethernet Address Resolution Protocol, or Converting Network Protocol Addresses to 48-bit Ethernet Addresses for Transmission on Ethernet Hardware," RFC 826.

This article is taken from the forthcoming book by Douglas Comer:

Internetworking With TCP/IP Principles, Protocols, and Architecture, Prentice-Hall, 1987

DOUGLAS COMER joined the faculty of the Computer Science Department at Purdue University after receiving a Ph.D. in Computer Science from Pennsylvania State University in 1976. He was principal investigator on the CSNET project, where he developed X25NET software. In addition to writing numerous papers on operating systems and networks, he developed the Xinu operating system and wrote two text books on operating system design. One of the books was written while Comer was on leave at Bell Laboratories. He is currently principal investigator of several network research projects, including the Cypress project, a member of the CSNET Executive Committee, and chairman of the CSNET Technical Committee, member of the Internet Activities Board (IAB), chairman of the the Distributed Systems Architecture Board (DSAB), and editor for the journal *Software Practice and Experience.*

An Overview of the Internet Activities Board

by Jon Postel, USC Information Sciences Institute

The Internet Activities Board (IAB) is the coordinating committee for the Internet. It is an independent committee of researchers and representatives of the sponsoring agencies. The membership changes over time to adjust to the current realities of research interests, development issues, Internet usage, and sponsorship.

IAB membership

The bulk of the members are the chairs of *Task Forces* that are chartered to work on specific aspects of the Internet. Membership on the task forces is determined by the chair, and is generally open to anyone that has the interest, time and resources to fully participate. Each task force may have an associated interest group open those who want to follow the work of the task force but are not able to be full participants. Some task forces are subdivided into *Working Groups* for work on specific topics or for short term studies. As the Internet evolves new task forces may be added and old task forces may be deleted.

The current membership of the IAB is:

Dave Clark	MIT	Chair	The Internet Architect
Jon Postel	ISI	ViceChair	The Deputy Internet Architect

Task Forces:

Bob Braden	ISI	End To End Services
Vint Cerf	NRI	Interagency Internet Management
Deborah Estrin	USC	Autonomous Networks
Phill Gross	Mitre	Internet Engineering
Steve Kent	BBN	Privacy and Security
Keith Lantz	Olivetti	Applications and User Interface
Barry Leiner	RIACS	Scientific Requirements
Jim Mathis	Apple	Robustness and Survivability
Dave Mills	UDEL	Internet Architecture

Related Activity and Sponsor Representatives:

Bill Bostwick	DoE	Department of Energy
Doug Comer	Purdue	Distributed Systems Activities Board Chair
Mike Corrigan	DoD	OSD-C^{3}I
Dave Farber	UDEL	NSF Networking Program Advisory Group Chair
Dennis Perry	DARPA	Internet Program Manager*
Steve Wolff	NSF	Director, NSF Division of Networking and Communications Research and Infrastructure
Tony Villasenor	NASA	Program Manager, Information Systems

(*Note: There is a transition in progress at DARPA-ISTO in the management of the Internet Program. It has not been decided at the time of this writing who will be participating in the IAB for DARPA.)

In the following the area of work of each task force is described briefly. The sizes of the task forces vary. Most of the task forces have five to twenty members, but the Engineering Task Force has well over 150 members.

End-to-End:
Bob Braden

The End-to-End Services Task Force is generally concerned with host-to-host communication services and protocols in the Internet. In the terminology of the OSI model, the task force interests falls into the transport, session, presentation, and application layers; in the Internet world, this generally includes all protocols above IP. The task force is also concerned with the service model that the internetwork sublayer (IP) provides to the transport layer.

Major areas of work by the task force include transaction protocols, remote procedure calls, multicasting, transport protocol performance, and structured data representations. The End-to-End Services Task Force has played an active role in encouraging the development of experimental prototypes of new protocols, especially IP multicasting, VMTP (transaction protocol), and NETBLT (high-performance data transfer).

Interagency Internet Management:
Vint Cerf

The Interagency Internet Management Task Force is a group convened to prepare recommendations for a research program over the next 3-5 years that would have the objective of extending the lifetime of the TCP/IP protocol suite until about 1997 when it is hoped that a full transition to the ISO protocol suite may be effected.

The group has prepared a report (currently in the editing stages). Once this report is accepted by the IAB, it is anticipated that this task force will disband unless it is requested by the IAB to examine other issues.

Autonomous Networks:
Deborah Estrin

The Task Force on Autonomous Networks is concerned with communication across the boundaries of computer networks that are autonomously developed, owned, and operated. An Autonomous Network (AN) is therefore a collection of gateways and hosts that is administratively autonomous from the rest of the world. Research in ANs addresses two related classes of problems:

1. Interconnection of two or more existing networks that are used and operated by separate administrations.

2. Decomposition of an existing network whose scale and scope make it difficult to modify and manage as a single, homogeneous network.

Both situations raise issues related to technical and administrative heterogeneity. Currently the task force is focusing on issues of administrative heterogeneity due to multiple and diverse administrations.

continued on next page

Overview of the Internet Activities Board *(continued)*

The task force is considering the design of AN gateways and internet protocols that will respect and enforce administrative boundaries and related access control requirements. To implement meaningful access controls, an autonomous network must be able to authenticate communications from outside its AN boundary.

Given the changes in environment and operating conditions that result from AN interconnection, the task force is reevaluating the functional requirements for gateways, network control protocols, and communication related applications (routing information distribution protocols, broadcast protocols, and application protocols such as directories).

Internet Engineering: *Phill Gross*

The Internet has grown to encompass a large number of widely geographically dispersed networks in academic and research communities. It now provides an infrastructure for a broad community of interest. Moreover, the family of Internet protocols and system components has moved from experimental to commercial development. To help coordinate the operation and management of the Internet, the IAB established the Internet Engineering Task Force with the charter to:

1. Act as a clearinghouse to promote the exchange of information within the Internet community. This community includes Internet software and hardware developers, Internet operators and Internet research and development groups.

2. Identify pressing and relevant short- to mid-range problem areas and convene Working Groups to explore solutions. Working Groups might deal with a wide range of Internet issues, such as operational management problems or protocol enhancements that would improve Internet performance.

3. Report Working Group results to the IAB and to the Internet community at large.

4. Recommend specific research and development projects aimed at solving operational problems.

Specific Working Groups are meant to have a fairly narrow scope and fixed duration. There are currently several Working Groups in various stages of progress focusing on Internet routing, performance, network management and network operations. For more information on the Internet Engineering Task Force, on attending meetings and proposing or joining Working Groups, contact Phill Gross, Mitre Corporation, (gross@gateway.mitre.org or 703-883-6794).

Privacy and Security: *Steve Kent*

The Privacy and Security Task Force addresses issues such as access control, authentication, restricted routing and privacy (non-disclosure) in the Internet environment. These topics are considered at various layers in the communication hierarchy, but with emphasis on the network, transport and application layers.

On some occasions, topics under discussion may require cleared personnel and thus a subgroup of the task force is comprised of individuals who will meet separately to participate in such discussions. In general, however, this task force will address issues in the context of communication and processing of unclassified information.

The task force began work by developing a list of privacy issues for further study, a list that is expected to grow over time with contributions from members of this and other task forces. The initial topic addressed by the task force was provision of privacy services for Internet text electronic mail. This work produced a document that details security mechanisms for providing confidentiality, integrity and authenticity for mail. Another document, currently in preparation, will provide specifications for key management.

Applications and User Interface: *Keith Lantz*

The Applications and User Interface Task Force investigates the requirements and makes constructive proposals for improved user interfaces to distributed computing environments. Features of distributed environments that distinguish them from non-distributed environments and impact the design and implementation of user interfaces include:

- additional constraints on delay and bandwidth between applications and users;
- increased heterogeneity -- of both hardware and software;
- new opportunities for loosely-coupled parallel processing; and
- the need to support collaboration between geographically distributed participants.

The task force develops requirements definitions and implementation strategies for these and related issues, keeping aesthetic concerns in mind. Current efforts are focused in two inter-related problem areas: user interface architecture and computer-supported, real-time, multi-media teleconferencing.

Scientific Requirements: *Barry Leiner*

The Task Force on Scientific Requirements is identifying the required networking technologies to support future scientific research. The intent is to identify the uses that scientists will make of networks, and to identify the technological requirements to support those uses. These requirements are passed on to other task forces to help guide and focus the networking research activities.

Examples of the issues being discussed are modes of interaction between workstations and supercomputers, the requirements for high bandwidth networking, requirements on electronic text and multi-media mail and conferencing facilities, and the requirements for privacy, access control and authentication.

continued on next page

Overview of the Internet Activities Board *(continued)*

An important topic of ongoing discussion is the interconnection of various agency and institution networks into a high performance system providing communications amongst the broad spectrum of scientific researchers and their required resources, both nationally and internationally.

The task force functions as an information exchange forum for multiple agency and research organizations to allow discussion and common formulation of requirements and suggested approaches in these areas.

Robustness and Survivability: *Jim Mathis*

The Robustness and Survivability Task Force considers the architectural and engineering issues of providing reliable, robust and survivable internetwork services for both commercial and military environments. Previous efforts include monitoring research, extending IP to provide the addressing and routing flexibility required to handle network partitions, network merges, internetwork portable hosts, and other forms of internetwork topology dynamics.

Current areas of research interest include advanced gateway algorithms that can improve internetwork survivability, impacts of survivable network technology on the internet architecture, fault isolation and diagnostic tools to increase reliability, and general protocol architectural problems in providing robust communication services.

Internet Architecture: *Dave Mills*

The Internet Architecture Task Force (INARC) studies technical issues in the evolution of the Internet from its present architectural model to new models appropriate for very large, very fast internets of the future. It is organized as a recurring workshop where researchers, designers and implementors can discuss novel ideas and experiences without limitation to the architecture and engineering of the present Internet. The output of this effort represents advance planning for a next-generation internet, as well as fresh insights into the problems of the current one.

For example, the INARC will hold a two-day retreat/workshop in December 1987 to discuss a fresh start on advanced internet concepts and issues. The agenda for this meeting is to explore architecture and engineering issues in the design of a next-generation internet system. There are invited presentations on selected topics followed by a general discussion on related issues. Written material from the workshop will be submitted for publication in the ACM Computer Communication Review.

Distributed Systems Activities Board: *Doug Comer*

The Distributed Systems Architecture Board (DSAB) is an organization analogous in structure to the IAB. Some task forces are joint with the two organizations. The purpose of the DSAB is to explore through experimental research the services and facilities needed for distributed computing and to make the results available to others.

It concentrates efforts on the services supplied by operating systems, the interface that the operating system supplies to application processes, and the interface between the operating system and underlying network modules that handle communication.

The DSAB is an independent organization, currently sponsored by the National Science Foundation (NSF) and the Defense Advanced Projects Research Agency (DARPA).

Networking Program Advisory Group: *Dave Farber*

The Networking Program Advisory Group (NPAG) and its predecessor, the NPAC, have been a fundamental part of the support and advisory mechanisms for the NSF's Division of Networking and Communications Research and Infrastructure (DNCRI). The NPAG is composed of communications, networking, and management experts from the academic community and from industrial laboratories. It is composed of four active working committees: the Research Committee (RC); the Technical Committee (TC), chaired by Hans-Werner Braun; Plans and Policy (PP), Larry Landweber; and Management (MC), Tony Villasenor.

The TC has made major contributions to the NSFNET through its initial concept documents, which laid the foundation for the overall architecture of the network; and through its ongoing oversight and technical assistance activities. It has been an important source of recommendations and support to the network managers during the initial phases of the deployment of NSFNET.

Recently the Management Committee was responsible for the structure and overall contents of the Backbone Management Solicitation for NSFNET, the Research Committee advised on the Division's recently-released Program Announcement for the Networking and Communications Research program, while the Plans and Policy Committee has been energetic in providing input to NSF management on network access policy, regional structure, and other critical issues.

Getting involved

New participants in the program of the IAB are always welcome. Please contact Jon Postel at POSTEL@ISI.EDU or 213-822-1511 if you would like to join the in the work of the IAB by participating on a task force or working group.

JONATHAN B. POSTEL is Associate Director of the Systems Division of the Information Sciences Institute of the University of Southern California. Jon has been involved in the development of computer communication protocols and applications from the early days of the ARPANET. His current interests include multimachine internetwork applications, multimedia conferencing and electronic mail, very large networks, and very high speed communications. Jon received a BS and MS in Engineering and a PhD in Computer Science from the University of California, Los Angeles.

'Twas the Night Before Start-up'

by Vint Cerf

Twas the night before start-up and all through the net,
not a packet was moving; no bit nor octet.
The engineers rattled their cards in despair,
hoping a bad chip would blow with a flare.
The salesmen were nestled all snug in their beds,
while visions of data nets danced in their heads.
And I with my datascope tracings and dumps
prepared for some pretty bad bruises and lumps.
When out in the hall there arose such a clatter,
I sprang from my desk to see what was the matter.

There stood at the threshold with PC in tow,
An ARPANET hacker, all ready to go.
I could see from the creases that covered his brow,
he'd conquer the crisis confronting him now.
More rapid than eagles, he checked each alarm
and scrutinized each for its potential harm.

On LAPB, on OSI, X.25!
TCP, SNA, V.35!

His eyes were afire with the strength of his gaze;
no bug could hide long; not for hours or days.
A wink of his eye and a twitch of his head,
soon gave me to know I had little to dread.
He spoke not a word, but went straight to his work,
fixing a net that had gone plumb berserk;
And laying a finger on one suspect line,
he entered a patch and the net came up fine!

The packets flowed neatly and protocols matched;
the hosts interfaced and shift-registers latched.
He tested the system from Gateway to PAD;
not one bit was dropped; no checksum was bad.
At last he was finished and wearily sighed
and turned to explain why the system had died.
I twisted my fingers and counted to ten;
an off-by-one index had done it again...

Multi-Vendor Support

Here is the most recent list of TCP/IP vendors. Over 160 companies now offer products which support TCP/IP and more are joining every day. If you know of other companies that should be added to the list please let us know.

ACC
Adax
Advintech
Alliant
Altos
AMD
Amdahl
Apollo Computer
Apple Computer
Applitek
Arete
ARINC Research
Associated Computer Experts
AT&T
Auscom
Aydin Monitor Systems
Banyan
BBN Communications
BBN Labs
BDM
Beame & Whiteside Software Ltd.
Bridge Communications
Britton Lee
Bull AG
Butler & Curless
Canaan Computer
Celerity Computing
Charles River
cisco Systems
Claflin & Clayton
Communication Machinery Corp.
Computer Network Technology
Concurrent Computing
Control Data Corporation
Convergent Technologies
Convex Computer
Computer Science Corporation
Counterpoint
Cray Research
Cydrome
Dana Group
DANET GmbH
Data General
Datapoint
DevelCon
Digital Equipment Corporation
Eastman Communications
Elxsi
Encore
Epilogue Technology
EXCELAN
Fibronics-Spartacus
Ford Aerospace
Fortune
Frontier Technologies
FTP Software Inc.
GE Calma
GEI Rechnersysteme GmbH
General Electric
Gold Hill Computers
Gould
GTE
Harris
Hemispheres Hi-Tech
Hewlett-Packard
Hirschmann
Honeywell
IBM
Imagen
Informix
Intel
Interactive Systems
Interfirm Graphic Systems Inc.
Integrated Solutions
Intergraph
Intermetrics
Internet Systems Corp.
Kinetics
KMW Systems
Lachman Associates
LANEX
Little Machine Inc.
Locus
Los Alamos National Labs
Marble Associates Inc.
MARI
Masscomp
Maxim
MICOM-Interlan
Microport
Mips
Mitek
Mitre
Motorola
Mt Xinu
NBI
NCR
NCR Comten
Network General Corp.
Network Research Corp.
Network Solutions
Network Strategies
Network Systems Corporation
Nixdorf
Norsk Data A/S
Novell
Opus
Oracle
Panda Programming
PCS Computer Systeme GmbH
Phoenix Technology
Plexus
Prime
Process Software
Proteon
Protocom Devices
Pyramid
Relational Technology
Research Equipment Inc.
Ridge Computer
SAIC
Santa Cruz Operation
SCI Inc.
Schlumberger/Applicon
Scope
Sequent
SIEMENS
Silicon Graphics
SMS Data Products Group
Software Kinetics Ltd.
Software Systems Associates
SPARTA
Spider Systems Inc.
SRI International
Stemmer Elektronik
Sterling Software
Stollmann GmbH
Stride Micro
Sun Microsystems
SYNELEC Datensysteme GmbH
Symbolics
Sytek
Tandem
Telemation
Televideo
Tektronix
Texas Instruments
THG
TOPS
3Com
Tracor
Ungermann-Bass
Uniq Digital Technologies
Unisoft
Unisys
Univation
Valid Logic
Vitalink
Wang
Wellfleet
Wetronic Automation GmbH
The Wollongong Group
Xerox
Xios Systems

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040

CONNEXIONS

PUBLISHER Daniel C. Lynch

EDITOR Ole J. Jacobsen